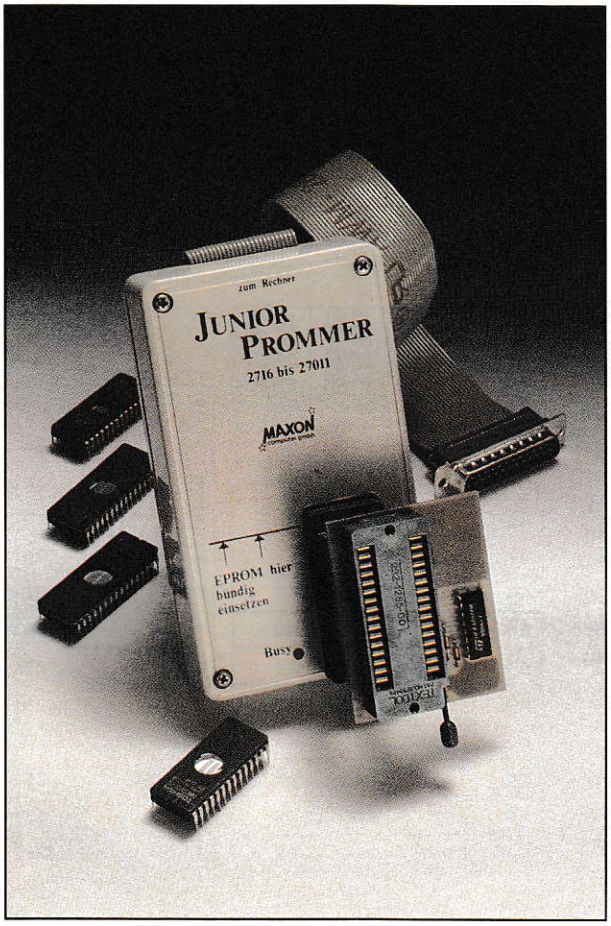
JUNIOR-PROMMER

Der EPROM-Brenner



Bei diesem Hardware-Projekt handelt es sich keineswegs um einen weiteren Eprommer der absoluten Billigklasse oder der Nobelklasse, sondern vielmehr um ein extrem günstiges aber auch sehr leistungsfähiges Gerät. Sie haben hier die Möglichkeit, die genaue Funktionsweise des Junior Prommers und damit die Funktionsweise von Eprommern schlechthin kennenzulernen.

Der Junior Prommer ist weder der erste seiner Art noch wird er der letzte sein, doch ist er wohl für viele, die eine Ausgabe von über 200 DM scheuen, nur um mal so'n Käfer zu brennen, eine ernsthafte Alternative. Mit einem Materialaufwand von etwa 70 DM und ein wenig Geduld können auch Sie ein Gerät besitzen, das folgendes leistet:

1. Programmiert alle Eproms vom Typ 2716 bis zum Typ 27011 und ist durch flexible Auslegung auch für zukünftige, 28polige Speicherbausteine geeignet. Mit dem Mega-Adapter (siehe Kasten 'Mega-Adapter') können sogar die 32poligen EPROMs 271001 bis 271008 gebrannt werden.

2. Vollkommen softwaregesteuert, wodurch ernsthafte Fehlbedienungen weitgehend ausgeschlossen werden.

3. Betrieb mit einer einzigen Spannung von 5V bei einem maximalen Strombedarf von 400 mA. Bei einem Anschluß am Joystickport (siehe auch Artikel über den Druckerport-Expander) wird kein externes Netzteil benötigt.

4. Kann an jeder normgerechten Centronics-Schnittstelle (z.B. auch an der des AMIGAs oder des IBM-PCs) betrieben werden.

5. Bietet die Möglichkeit festzustellen, ob ein Speicherbaustein eingesetzt ist.

6. Bearbeitet außerdem noch ROMs, EEPROMs und pinkompatible Speicherbausteine.

Was sind eigentlich EPROMs?

Nüchtern gesehen sind es schlicht Speicherbausteine, die zu jeder ihnen angelegten Eingangsadresse ein passendes und programmierbares Ausgangsdatenwort parat haben. Der Wertebereich der Eingangsadressen zusammen mit dem der Ausgangsdatenwörter ist ein Maß für die Speicherkapazität. Heute gebräuchliche EPROMs sind die der 27XXX-Serie (Tabelle 1). XXX steht fast immer für die Speicherkapazität in Kilobit (Bit/1024). Oft eingesetzt als Programmspeicher in Microcomputern, hat das EPROM einen Nachteil und einen entscheidenden Vorteil:

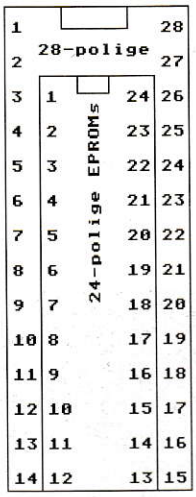
Programmiert werden muß ein EPROM mit einer recht aufwendigen und zeitraubenden Prozedur, wozu es natürlich EPROMmer gibt. Jedoch bleiben die einmal programmierten Daten trotz fehlender Versorgungsspannung bis zum Löschen durch starkes ultraviolettes Licht erhalten.

Jetzt ist auch der Name verständlicher: EPROM ist die Abkürzung für Erasable-Programmable-Read-Only-Memory. Zu deutsch: löschbarer und programmierbarer Festwertspeicher.

Da ein EPROM mit UV-Licht gelöscht werden soll, ist es meist mit einem Quarzglasfenster ausgestattet, durch das man den Halbleiterkristall mit UV-Licht bestrahlen kann, um alle Speicherzellen in einen definierten (gelöschten) Ausgangszustand zu bringen. Daher sollte es selbstverständlich sein, daß ein EPROM durch das Aufkleben eines UV-lichtdichten Aufklebers vor ungewolltem Löschen durch den in normalem Umgebungslicht enthaltenen UV-Anteil bewahrt wird.

Intern werden die digitalen Informationen durch das Vorhandensein oder Nichtvorhandensein einer Ladung gespeichert. Das Aufbringen dieser Ladung geschieht auf elektrischem Wege, eben durch das Programmieren mit dem Junior Prommer, das Löschen der Ladung jedoch nur über UV-Licht.

27011 128 kB	27513 64 kB	27512 64 kB	27256 32 kB	27128 16 kB	2764 8 kB	2732 4 kB	2716 2 kB
Vpp/$\overline{RST}$	$\overline{RST}$	A15	Vpp	Vpp	Vpp	----	----
A12	A12	A12	A12	A12	A12	----	----
A7	A7	A7	A7	A7	A7	A7	A7
A6	A6	A6	A6	A6	A6	A6	A6
A5	A5	A5	A5	A5	A5	A5	A5
A4	A4	A4	A4	A4	A4	A4	A4
A3	A3	A3	A3	A3	A3	A3	A3
A2	A2	A2	A2	A2	A2	A2	A2
A1	A1	A1	A1	A1	A1	A1	A1
A0	A0	A0	A0	A0	A0	A0	A0
D0	D0	D0	D0	D0	D0	D0	D0
D1	D1	D1	D1	D1	D1	D1	D1
D2	D2	D2	D2	D2	D2	D2	D2
Gnd	Gnd	Gnd	Gnd	Gnd	Gnd	Gnd	Gnd

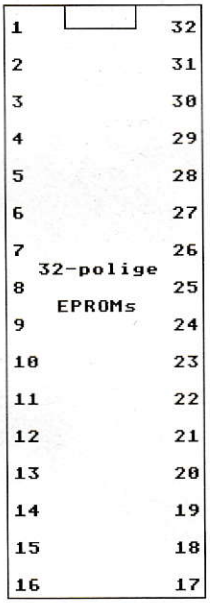


2716 2 kB	2732 4 kB	2764 8 kB	27128 16 kB	27256 32 kB	27512 64 kB	27513 64 kB	27011 128 kB
----	----	+5V	+5V	+5V	+5V	+5V	+5V
----	----	$\overline{PGM}$	$\overline{PGM}$	A14	A14	$\overline{WE}$	$\overline{PGM}$/$\overline{WE}$
+5V	+5V	A13	A13	A13	A13	A13	A13
A8	A8	A8	A8	A8	A8	A8	A8
A9	A9	A9	A9	A9	A9	A9	A9
Vpp	A11	A11	A11	A11	A11	A11	A11
$\overline{OE}$	$\overline{OE}$/Vpp	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$/Vpp	$\overline{OE}$/Vpp	$\overline{OE}$
A10	A10	A10	A10	A10	A10	A10	A10
$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
D7	D7	D7	D7	D7	D7	D7	D7
D6	D6	D6	D6	D6	D6	D6	D6
D5	D5	D5	D5	D5	D5	D5	D5
D4	D4	D4	D4	D4	D4	D4	D4
D3	D3	D3	D3	D3	D3	D3	D3

Tabelle 1a. Alle EPROM-Typen, die mit dem Junior Prommer ohne den Mega-Adapter bearbeitet werden können

Heute sind bereits EPROMs mit Kapazitäten bis zu einem Megabit (z.B. 27011) gängig. Da fast alle EPROMs eine Datenbreite von 8 Bit haben, muß man nur die Zahl hinter der 27 durch 8 teilen, um die Kapazität in Kilobyte zu erhalten. Eine Ausnahme bilden hier 27011 und 27513, mit denen es jedoch etwas besonderes auf sich hat. Beide sind sogenannte 'gebankte' EPROMs, was heißt, daß sie mangels Adreßleitungen in mehrere Abschnitte (Banks) aufgeteilt wurden, die dann mit einer speziellen Prozedur ausgewählt werden. Daher hat man zwar nicht mehr an Kapazität parallel verfügbar, jedoch kann man aus einem großen Bereich wählen. Der 27513 ist organisiert zu vier Bereichen je 16kByte. Der 27011 ist organisiert zu 8 Bereichen je 16kByte. Alle zukünftigen EPROMs höherer Kapazität im 28poligen Gehäuse müssen zwangsläufig auf demselben Ansteuerprinzip beruhen.

278001 1 MB	274001 512 kB	272001 256 kB	271001 128 kB
A19	Vpp	Vpp	Vpp
A16	A16	A16	A16
A15	A15	A15	A15
A12	A12	A12	A12
A7	A7	A7	A7
A6	A6	A6	A6
A5	A5	A5	A5
A4	A4	A4	A4
A3	A3	A3	A3
A2	A2	A2	A2
A1	A1	A1	A1
A0	A0	A0	A0
D0	D0	D0	D0
D1	D1	D1	D1
D2	D2	D2	D2
Gnd	Gnd	Gnd	Gnd



271001 128 kB	272001 256 kB	274001 512 kB	278001 1 MB
+5V	+5V	+5V	+5V
PGM	PGM	A18	A18
----	A17	A17	A17
A14	A14	A14	A14
A13	A13	A13	A13
A8	A8	A8	A8
A9	A9	A9	A9
A11	A11	A11	A11
$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$/Vpp
A10	A10	A10	A10
$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
D7	D7	D7	D7
D6	D6	D6	D6
D5	D5	D5	D5
D4	D4	D4	D4
D3	D3	D3	D3

Tabelle 1b. Die 32poligen EPROM-Typen (Bearbeitung nur mit dem Mega-Adapter und entsprechender Software möglich!!!)

Ansteuerung der EPROMs

Das Lesen der Daten aus einem EPROM ist trivial, und außerdem gleicht es hier fast allen anderen Speichern, die der Jedec-Norm folgen, wie z.B. ROMs, Statische RAMs etc. Sobald stabile Eingangssignale an den Adreßleitungen anliegen und sowohl /CE (/ = Negation, d.h. das Signal ist dann aktiv, wenn es low ist) als auch /OE aktiv sind, liegt auch nach einer gewissen Verzögerung, genannt Zugriffszeit, das über die Adresse gewählte 8-Bit-Ausgangsdatenwort an. Die beiden Steuersignale /CE und /OE haben die Aufgabe, das EPROM zu selektieren und dessen Ausgangstreiber zu kontrollieren. /CE = 1 schaltet das ganze EPROM ab, unabhängig vom /OE Signal. /OE kontrolliert im Falle eines eingeschalteten EPROMs die Ausgangstreiber (low-aktiv). Die restlichen Steuerleitungen haben nur in Verbindung mit der Programmierung eine Bedeutung.

Programmierung von EPROMs

Das Programmieren läuft generell bei allen EPROMs ähnlich ab. Nachdem Adressen, Daten und eventuell auch die Programmierspannung stabil anliegen, wird für eine festgelegte Zeit der Programmierimpuls gegeben. Danach legt man neue Daten und Adressen an und wiederholt den Programmierimpuls, bis alle Adressen programmiert sind. Wie das genau vor sich geht, hängt vom Typ und manchmal sogar vom Hersteller ab. Der Programmieralgorithmus bestimmt, welcher Weg eingeschlagen wird, um die Daten dauerhaft zu programmieren. Hier gibt es grob zusammengefaßt drei Möglichkeiten:

1. Standardalgorithmus: das Programmieren mit einer konstanten Zeit (50ms), nach der angenommen wird, daß die Daten sicher gespeichert sind (nicht sehr effektiv).

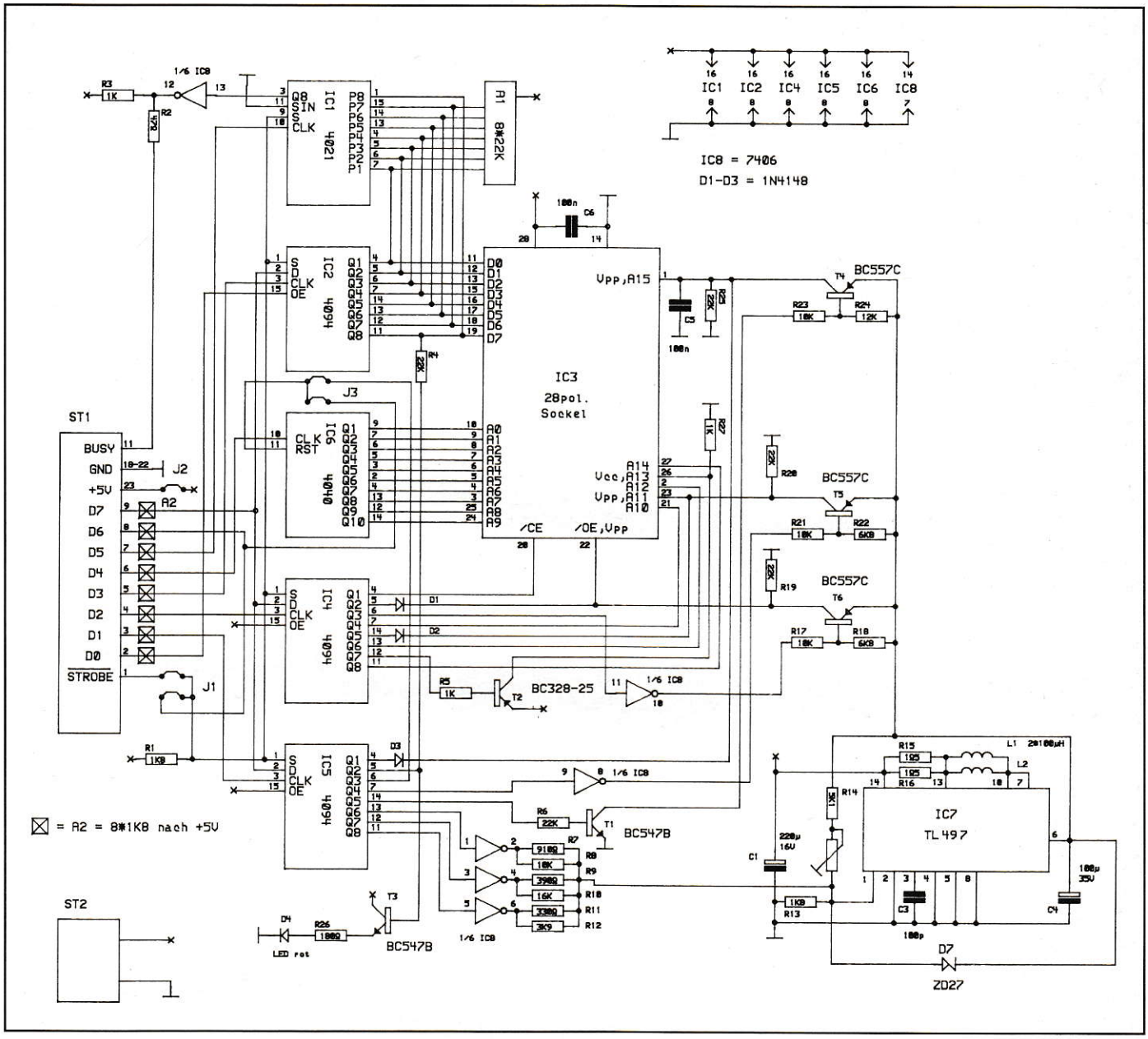


Bild 2a. Das Schaltbild des Junior Prommers. J1 bis J3 dienen einer Anpassung speziell an AMIGA und andere Rechner

2. Bei 'intelligenten Algorithmen' arbeitet man mit wesentlich kürzeren Zeiten (1mS), nach denen jeweils geprüft wird, ob die Daten bereits akzeptiert wurden. Danach wird noch einmal mit einem Vielfachen (3 bis 4) der davor benötigten Zeit darüberprogrammiert, um eine sichere Programmierung aller Zellen zu gewährleisten. Die erreichbaren Zeiten liegen etwa um ein Zehnfaches unter denen der Methode 1, jedoch wird sie erst ab dem Typ 2764 aufwärts unterstützt.

3. Beim sogenannten Quickpulse-Algorithmus ist der Vorgang fast identisch mit dem von Punkt 2, jedoch arbeitet man mit noch kürzeren Programmierimpulsen (0.1ms) und übergeht das Nachprogrammieren. Dies ist zweifellos die schnellste Methode, doch wird sie nur ab dem Typ 27512 aufwärts unter speziellen Bedingungen empfohlen.

Der Standardalgorithmus kann generell auf jedes EPROM angewendet werden. In Abbildung 3 und 4 kann man einen Überblick über alle Speichertypen gewinnen. Weiterhin kann man eine EPROM-Speicherzelle zwar elektrisch programmieren, jedoch kann man ein Bit nur elektrisch auf 0 bringen und nur durch UV-Licht wieder in den 'gelöschten' 1-Zustand.

Um die Implementierung neuer EPROMs in bestehende Systeme zu vereinfachen, geben Hersteller in ihren Datenblättern meistens eine Tabelle an, aus der entnommen werden kann, welche Signale man für jeden spezifischen Betriebszustand an die Steuerleitungen des EPROMs legen muß. Falls Sie daran interessiert sind, sollten Sie auf jeden Fall die Datenblätter der Hersteller zu Rate ziehen. Deshalb soll die Programmierung hier nur an dem konkreten Beispiel des Typs 2764 erläutert werden.

Beispiel Typ 2764

Zuerst muß das EPROM gelöscht sein, d.h. alle Bits haben den 1-Zustand, bevor man sie eventuell auf 0 programmieren kann. Nachdem die Versorgungsspannung eingeschwungen ist und alle Steuersignale inaktiviert sind, wird die Programmierspannung von hier 21V an den Pin 1 (Vpp) gelegt, um das EPROM in den Programmiermodus zu bringen. Jetzt wird der Pin /CE auf 0 gesetzt, um das EPROM zu aktivieren. Danach werden die erste zu programmierende Adresse und das Datenwort angelegt, um danach den Pin /PGM für eine Millisekunde auf 0 zu pulsen. Um die Richtigkeit der programmierten Daten zu prüfen, werden die Ausgangstreiber über /OE aktiviert und mit dem programmierten Datenwort verglichen. War dieser Schritt nicht erfolgreich, so kann er bis zu 15mal wiederholt werden,

Widerstände:
alle 1/4 Watt Kohleschicht

R1,R13 :	1K8
R2 :	47Ω
R3,R5,R27 :	1K
R4,R6,R19, R20,R25 :	22K
R7 :	910Ω 1% Metallschicht
R8 :	10K 1% Metallschicht
R9 :	390Ω 1% Metallschicht
R10 :	16K 1% Metallschicht
R11 :	330Ω 1% Metallschicht
R12 :	3K9 1% Metallschicht
R14 :	5K1
R15,R16 :	1Ω5
R17,R21,R23 :	10K
R18,R22 :	6K8
R26 :	180Ω

Kondensatoren:

C1 :	220µ / 16V Elko, stehend
C3 :	100p keramisch
C4 :	100µ / 35V Elko, stehend
C5,C6 :	100n vielschicht

Halbleiter:

D1-D3 :	1N4148
D4 :	LED 3mmØ rot
D5 :	Zenerdiode 33V / 400mW
T1,T3 :	BC547B
T2 :	BC328-25
T4-T6 :	BC557B
IC1 :	4021
IC2,IC4,IC5 :	4094
IC3 :	28 pol. Nullkraftsockel
IC6 :	4040
IC7 :	TL497
IC8 :	7406

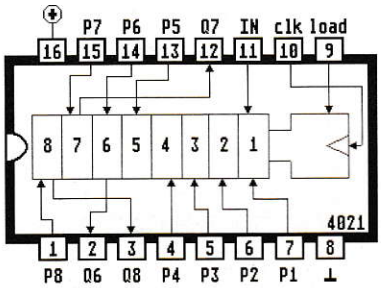
Sonstiges:

ST1 :	Doppelpfostenleiste, 26 polig
ST2 :	Stromversorgungsbuchse (Printmontage)
TR1 :	Trimmer klein, stehend 1K
A1 :	Single-Inline-Array 8*22K
A2 :	Single-Inline-Array 8*1K8

J1-J3 : je nach Rechnertyp
L1,L2 : Miniaturdrossel 100µH (Conrad) oder Netzentstördrossel mit 50µH
evtl. IC-Fassungen : 4*DIL 14, 5*DIL 16
25poliger Sub-D Quetschstecker
26poliger, doppelreihiger Pfostenverbinder, Quetschausführung
ca. 50cm 26 adriges Flachbandkabel
Gehäuse : 123L * 79B * 27T
doppelseitige durchkontaktierte Platine 'Junior Prommer'

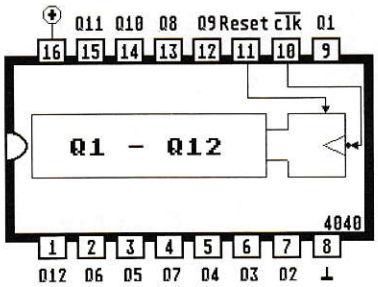
Bild 2b. Die Stückliste des Junior Prommers

Die verwendeten ICs



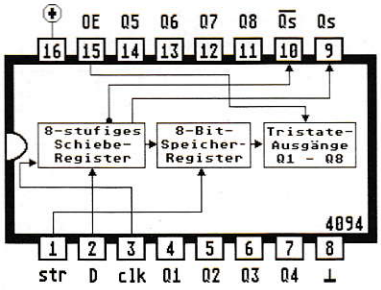
4021

Der C-MOS IC 4021 ist ein 8stufiges Schieberegister. Das an den Eingangspins (P1 bis P8) anliegende Daten-Byte wird, wenn Load High ist, in das Schieberegister übernommen. Nun können die einzelnen Bits des Daten-Bytes durch getaktetes (clk) Verschieben an den Ausgängen seriell, d.h. nacheinander, ausgelesen werden.



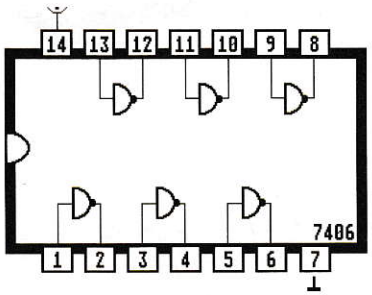
4040

Bei dem C-MOS-IC 4040 handelt es sich um einen 12stufigen Binär-Zähler. Wenn an Reset ein High-Signal anliegt, werden alle Ausgänge in ihren Ursprungszustand (Low) gesetzt, d.h. der Zähler wird auf 0 gestellt. Während des normalen Betriebs liegt dieser Eingang auf Low. Mit jedem High-Low-Übergang des Clock-Signals (clk) wird der Zähler um Eins erhöht. Der Zählerstand liegt zu jeder Zeit an den Ausgängen (Q1 bis Q12) an.



4094

Der IC besitzt einen seriellen Eingang (D). Bei jedem Clock-Impuls (clk) wird das dort anliegende Signal (Low oder High) in das Schieberegister übernommen und sein Inhalt weitergeschoben. An den seriellen Ausgängen (Qs und /Qs) steht das Signal zur Verfügung, das vor acht Clock-Impulsen eingelesen wurde. Wenn die Strobe-Leitung auf High liegt, werden die 8 Daten-Bits, welche sich im Schieberegister befinden, in das Speicherregister übernommen. Solange der Ausgangsfreigabeeingang (OE) auf Low liegt, sind alle parallelen Ausgänge hochohmig und stören den anschließenden Bus nicht. Wird OE auf High gelegt, erscheinen an den Ausgängen die Daten-Bits der Speicherregister.



7406

Der 7406 beinhaltet sechs Inverter. Im Gegensatz zum Standard-Inverter-Baustein (7404) hat dieser offene Kollektorausgänge.

bevor ein Fehler gemeldet wird. Aufgrund der verlangten Datensicherheit wird noch mit der vierfachen der bisher benötigten Zeit nachprogrammiert, bevor zur nächsten Speicherstelle übergegangen wird. Zuletzt überprüft man das vollständige EPROM noch einmal auf die Richtigkeit der Daten, bevor man es über /CE inaktiviert und die Programmierspannung abschaltet.

Der Junior Prommer

Der Junior Prommer ist konzipiert für den Betrieb an jeder Standard-Centronics Schnittstelle. Er stellt alle nötigen Programmierspannungen und Digitalleitungen zur Verfügung. Um die zehn Leitungen dieser Schnittstelle auf die nötige Anzahl zu erweitern, wird ein serielles Übertragungsprinzip angewandt. Ausgangsleitungen (insgesamt 24) werden mit einem Baustein 4094 realisiert, bei dem es sich um ein 8-Bit-"seriell ein/parallel aus"-synchrones Schieberegister mit nachgeschaltetem Register und schaltbaren Ausgängen handelt (siehe IC-Kasten). Insgesamt drei dieser Schieberegister sind vorhanden, wobei alle Leitungen bis auf den TAKT parallel angesteuert werden, damit ein individueller Betrieb der Register möglich wird. Die Ausgänge von IC2 sind steuerbar, um einen bidirektionalen Datentransfer mit dem EPROM abwickeln zu können. Eingangsleitungen (insgesamt 8) werden mit dem Baustein IC1 / 4021 bedient, bei dem es sich um ein "parallel ein/seriell aus"-Schieberegister handelt, das wiederum mit separatem Takt betrieben wird (siehe IC-Kasten). Um die Schnittstelle nicht modifizieren zu müssen, werden die seriellen Daten von IC1 über den Pin BUSY in den Rechner geleitet. Zusätzlich existiert noch ein Zähler IC6/4040 (siehe IC-Kasten), der die unteren zehn Adreßleitungen bedient und über die Schnittstelle inkrementiert sowie gelöscht werden kann.

Die Programmierspannungen werden an Bord mit IC7/TL497, einem frequenzgesteuerten Sperrwandler (siehe IC-Kasten), erzeugt und über IC5 eingestellt auf 5V, 12.5V, 21V bzw. 25V. Der Sperrwandler ist in der Lage, auch bei höchster Programmierspannung den nötigen Strom von etwa 30mA zu liefern. Mit T4 bis T6 kann diese Programmierspannung dann wahlweise auf die Pins 22, 23 und 1 geschaltet werden. Gleichzeitig können diese Pins auch über digitale Ausgänge der Schieberegister kontrolliert werden. T2 schaltet bei 24poligen Speichern die Versorgungsspannung. Die Stromaufnahme des Junior Prommers ist ganz von den verwendeten EPROMs abhängig und liegt in der Regel unter 300mA. Die Versorgung des Junior Prommers kann der Rechner übernehmen, ist jedoch auch mit einem stabilisierten 5V-Netzteil über ST2 möglich.

Die Bedeutung der Steckbrücken

Mit dem Jumper J1 kann zwischen Betrieb am ST (/STROBE) und am AMIGA (D6) zur Freigabe der Register gewählt werden. J2 ermöglicht bei einem Betrieb am AMIGA die Stromversorgung über diesen; beim ST bleibt die Brücke offen. J3 ist wieder für den wechselweisen Betrieb am ST (Rücksetzten des Zählers über D6) und am AMIGA (durch Ausgang Q3 des IC 5).

Die Programmierung der Hardware

Bis auf die Pins 22, 23 und 1, auf die wahlweise eine Programmierspannung von 5V, 12.5V, 21V oder 25V geschaltet werden kann, werden alle Pins digital angesteuert. Eine Sonderstellung hat Pin 26, der zusätzlich den nötigen Strom liefert, um 24polige EPROMs zu versorgen.

Wie Sie schon wissen, wird der Datentransport zwischen Rechner und Easyprommer seriell über zehn Leitungen einer Centronics- Schnittstelle abgewickelt. In der Hauptsache besteht der Junior Prommer daher aus Schieberegistern und einem Zähler:

1. Ausgangsschieberegister SRA (IC5): Es steuert die Programmierspannung und deren Verteilung.

2. Ausgangsschieberegister SRB (IC4): Dessen Aufgabe ist es, die Steuerleitungen und oberen Adressen des EPROMs zu kontrollieren.

3. Ausgangsschieberegister SRC (IC2): In das werden die ins EPROM zu programmierenden Daten geschoben.

4. Eingangsschieberegister SRD (IC1): Aus ihm werden die Zustände der Datenleitungen des EPROMs geschoben.

5. Adreßzähler AZ (IC6): Er bedient die unteren Adreßleitungen.

Gängige Speichertypen der JEDEC-Norm:

EPROMs:

Typ	Kapazität	Programmier-Spannung	Programmier-Algorithmus
2716	2kByte	25V	50mS
2732	4kByte	25V	50mS
2732A	4kByte	21V	50ms
2764	8kByte	21V	1mS,4*
2764A	8kByte	12,5V	1mS,3*
27128	16kByte	21V	1mS,4*
27128A	16kByte	12,5V	1ms,3*
27256	32kByte	12,5V	1mS,3*
27512	64kByte	12,5V	1mS,3*/0,1mS
27513	16kByte*4	12,5V	1mS,3*/0,1ms
27011	16kByte*8	12,5V	1mS,3*/0,1ms
271001	128kByte	12,5V	
272001	256kByte	12,5V	
274001	512kByte	12,5V	
278001	1MByte	12,5V	

EEPROMs:

Typ	Kapazität	Programmier-Spannung	Programmier-Algorithmus
2804	0,5kByte	5V	10mS
2816	2kByte	5V	10mS
2864	8kByte	5V	10mS

Tabelle 2. Gängige Speichertypen der JEDEC-Norm

Die benutzten Leitungen und deren Bedeutung sollen jetzt aufgezählt werden:

/STROBE, Pin 1:

Normalerweise wird mit diesem Signal dem Drucker die Gültigkeit der Daten angezeigt. Ähnlich auch hier: Wird STROBE auf 1 gesetzt, werden alle Register im Easyprommer geladen. Der Ruhezustand dieser Leitung ist jedoch nicht 1, sondern 0. Das heißt, der Inhalt in den Schieberegistern der 4094 wird zu den Ausgängen übertragen, und in IC1 wird das Datenwort des EPROMs parallel geladen, um danach eventuell herausgeschoben zu werden.

D0, Pin 2:

Dieses Signal kontrolliert die Ausgangstreiber von SRC und damit die Datenrichtung zu oder von dem EPROM. Ist D0 = 1, gibt SRC seine Daten an die Datenleitungen des EPROMs. Falls D0 = 0 ist, können die Datenleitungen des EPROMs über SRD gelesen werden. Es ist darauf zu achten, daß kein gleichzeitiges Aktivieren der Ausgangstreiber des EPROMs und der des SRCs auftreten kann.

D1, Pin 3:

Dies ist die Taktleitung von IC5 (SRA),

der die Programmierspannungen und deren Verteilung steuert. Bei der ansteigenden Flanke von D1 wird das Bit an D7 in das Schieberegister des SRA geschoben.

D2, Pin 4:
Dies ist ebenfalls eine Taktleitung, jedoch für IC4(SRB), der die Steuersignale des EPROMs kontrolliert.

D3, Pin 5:
Die Taktleitung für IC2(SRC), der die Datenleitungen des EPROMs kontrolliert.

D4, Pin 6:
Die Taktleitung für den Adreßzähler (AZ). Die abfallende Flanke von D4 erhöht den AZ um 1. Der AZ bedient lediglich die Adressen A0 bis A9, um durch die Software-Kontrolle der übrigen Adressen eine höhere Flexibilität zu erreichen.

D5, Pin 7:
Die Taktleitung für IC1, über den die Datenleitungen des EPROMs gelesen werden können. Die jeweils ansteigende Flanke des Taktes bringt ein weiteres, invertiertes Bit aus dem Schieberegister IC1 an den Pin BUSY.

D6, Pin 8:
Rücksetzleitung des Adreßzählers IC6. Solange diese Leitung 1 ist, wird der Adreßzähler rückgesetzt.

D7, Pin 9:
Die gemeinsame Datenleitung aller Ausgangsschieberegister. Über diese Leitung wird der serielle Datenstrom zu den jeweils getakteten Ausgangsschieberegistern übertragen.

BUSY, Pin 11:
Über diesen Pin können durch IC1 die Datenleitungen des EPROMs seriell gelesen werden.

In der Praxis

Die Ansteuerung der Ausgangsschieberegister sollte mit einem Maschinenprogramm erfolgen, das den Centronics-Port direkt bedient. Das Schieben eines Datenwortes in eines der Ausgangsschieberegister folgt diesem Ablauf:

1. Das höchstwertige Bit des Datenbytes auf D7 legen.

2. Geben des Taktimpulses auf die jeweilige Taktleitung [D1 für SRA (IC 5), D2 für SRB (IC4) oder D3 für SRC (IC2)].

3. Einmaliges Linksschieben des Datenbytes

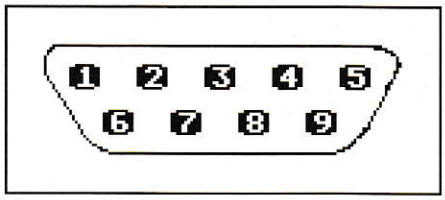


Bild 3. Die Joystick-Buchse von außen gesehen

4. Wiederholen ab 1, bis das Byte vollständig herausgeschoben wurde.

5. Geben des positiven STROBE-Impulses (/STROBE = 1), damit die Daten des jeweiligen Registers zu den Ausgängen transferiert werden.

Die Ansteuerung des Eingangsschieberegisters SRD differiert leicht:

1. Zuerst muß SRC über D0 inaktiviert und das EPROM aktiviert werden.

2. Geben des positiven STROBE-Impulses (/STROBE = 1), damit die Zustände der Datenleitungen in SRD übertragen werden.

3. Auf der BUSY-Leitung liegt nun das niederwertigste Bit der EPROM-Daten. Übernamhe des Datenbytes in den Rechner.

4. Linksschieben des Datenbytes.

5. Wiederholen ab 2, bis das vollständige Byte übertragen ist.

Die Bedeutung der Schieberegisterbits

SRA:

Bit 0 (Q1):
Steuert den Pin 1 des EPROMs digital an. Falls an Pin 1 schon eine Programmierspannung geschaltet ist, bleibt diese Leitung irrelevant. Zu beachten ist, daß - bedingt durch C5 - etwa 6 Millisekunden vergehen, bis der eindeutige Pegel am Sockel anliegt.

Bit 1 (Q2):
Hat zwei Funktionen. Zum einen steuert diese Leitung schlicht die Leuchtdiode D4 (1 = ein) und zum anderen kann man über sie die Treiberfähigkeit des EPROMs testen. Deswegen ist die Datenleitung D7 des EPROMs nicht mit einem 'Pull-up' verbunden, sondern über einen hochohmigen Widerstand mit dieser Leitung. Der Grund dafür: Ist ein EPROM eingesetzt und nicht in seiner Treiberfähigkeit gestört, ist der Zustand dieser Leitung nicht entscheidend für die gelesenen Daten. Ist jedoch kein EPROM eingesetzt oder dessen Treiberfähigkeit gestört, folgt das Bit 7 aus den gelesenen Daten dieser Leitung, woran festgestellt werden kann, ob kein EPROM eingesetzt ist, oder ob das EPROM einen Defekt hat. Während des normalen Junior Prommer-Betriebs sollte diese Leitung 1 sein.

Bit 2 (Q3):
Hier nicht belegt.

Bit 3 (Q4):
Schaltet die Programmierspannung auf Pin 23 des EPROMs (1 = Spannung ein).

Bit 4 (Q5):
Schaltet die Programmierspannung auf Pin 1 des EPROMs (1 = Spannung ein)

Bit 5-7:
Geben die Höhe der Programmierspannung nach folgender Zuordnung an:

000 = 5V
100 = 12.5V
010 = 21V
001 = 25V

Falls die Programmierspannung umgeschaltet wird, muß danach etwa eine Sekunde gewartet werden, bevor sie an einen Pin des EPROMs gelegt wird, da der Schaltregler sie sonst noch nicht ausreichend eingeregelt hat. Das gilt vor allem, wenn die Programmierspannung auf einen niedrigeren Wert geschaltet wird.

SRB:

Bit 0 (Q1):
Steuert direkt Pin 20 (/CE) des EPROMs.

Bit 1 (Q2):
Steuert digital Pin 22 (/OE,Vpp) des EPROMs an. Falls an diesem Pin schon Programmierspannung anliegt, ist dieses Bit irrelevant.

Bit 2 (Q3):
Schaltet die Programmierspannung auf Pin 22 des EPROMs. (1 = Spannung ein)

Bit 3 (Q4):
Steuert Pin 21 (A10) des EPROMs an.

Bit 4 (Q5):
Steuert digital Pin 23 (A11,Vpp) des EPROMs an. Wie immer hat die Programmierspannung Vorrang.

Bit 5 (Q6):
Steuert Pin 2 (A12) des EPROMs an.

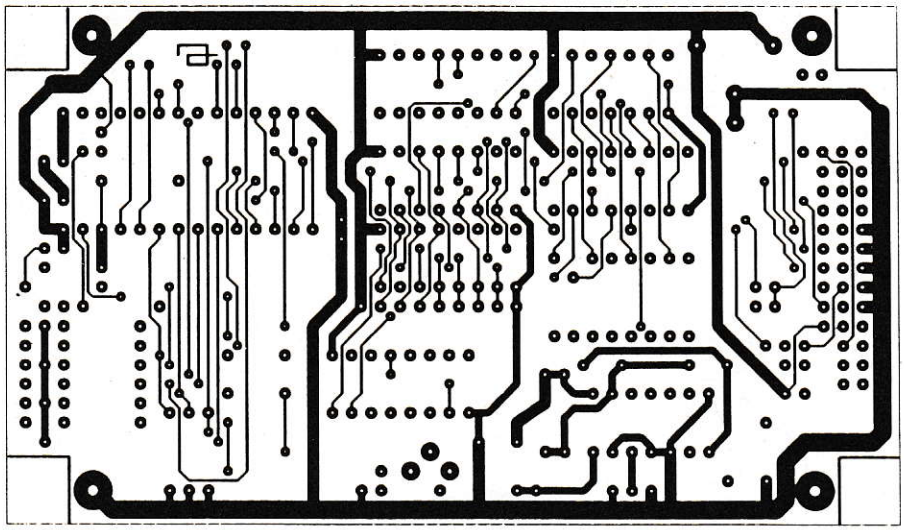


Bild 4a. Layout Lötseite

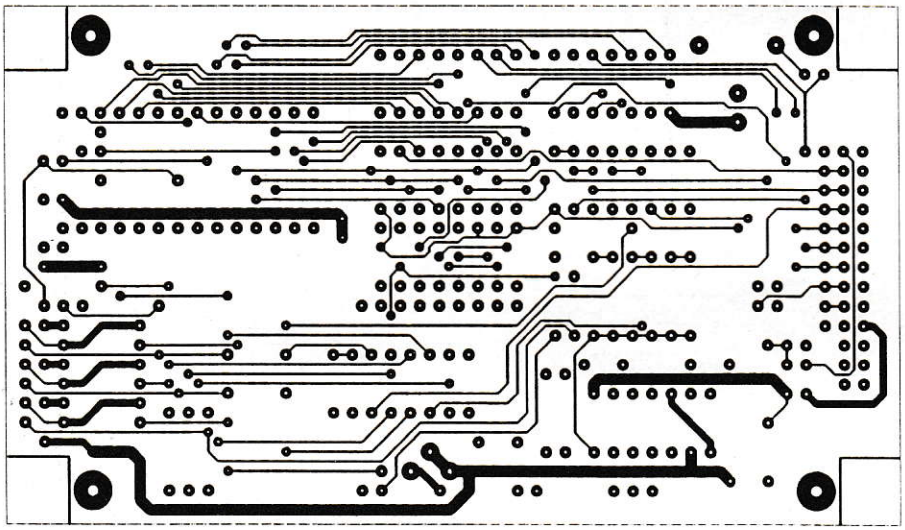


Bild 4b. Layout Bestückungsseite

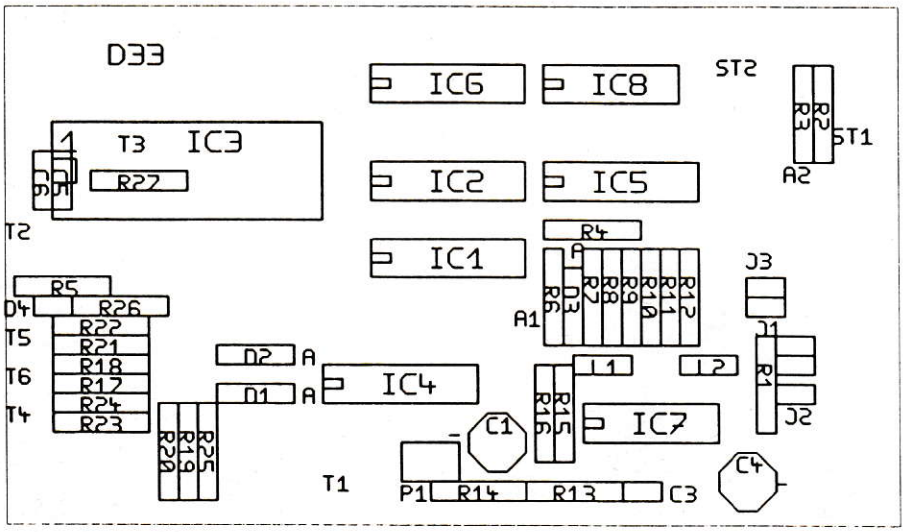


Bild 4c. Bestückungsaufdruck

Bit 6 (Q7):
Steuert invertiert Pin 26 (A13,Vcc) des EPROMs an. Diese Leitung liefert den entsprechenden Strom um gleichzeitig die Stromversorgung von 24poligen EPROMs zu übernehmen.

Bit 7 (Q8):
Steuert den Pin 27 (A14,/PGM) des EPROMs direkt an.

SRC und ebenfalls SRD sind direkt mit den entsprechenden Datenleitungen des EPROMs verbunden (siehe Schaltplan).

Der Aufbau

Da das Layout des Junior Prommers im Hinblick auf ein Gehäuse ausgelegt wurde, müssen der Nullkraftsockel (IC3) sowie die LED (D4) zuletzt auf die Lötseite montiert werden. Dadurch ist die Bauhöhe der anderen Bauteile, die sich ausschließlich auf der Bauteilseite befinden, unkritisch. Die Verbindung des Junior Prommers mit der jeweiligen Schnittstelle des Rechners geschieht über ein passend geschneidertes Flachbandkabel. Am einen Ende des unter 50cm langen, 26poligen Flachbandkabels wird ein 25poliger Sub-D-Stecker angequetscht und am anderen ein 26poliger Quetschverbinder. Natürlich muß Pin 1 des Sub-D-Steckers auch mit Pin 1 des Quetschverbinders zusammenliegen. Die 26. Ader des Flachbandkabels wird vor dem Sub-D-Stecker abgetrennt. Außerdem benötigen Sie ein Verbindungskabel zur 5V-Spannungsversorgung des Junior Prommers. Dabei können Sie entweder ein externes, stabilisiertes Netzteil verwenden, oder Sie greifen mit lediglich einer Leitung die Spannungsversorgung Ihres Rechners am Joystickport ab. Dazu benötigen Sie ein Verbindungskabel vom Joystickport Pin 7 (Bild 3) zum Mittelstift der Stromversorgungbuchse (ST2) des Junior Prommers. Selbstverständlich entfällt beim ST die Brücke (J2) zur Versorgung über die Schnittstelle. Jetzt ist der Junior Prommer, sofern fehlerfrei aufgebaut, betriebsbereit. Um die Funktionsfähigkeit zu prüfen, ist das Programm JUNIOR.BAS abgedruckt, mit dem die Hardware durchgecheckt werden kann.

Da das Layout stellenweise recht eng geraten ist, sollte man schon ein bißchen Lötfertigkeit besitzen.

Die Software

Über ein vielseitiges Assemblerprogramm, das Sie in beliebige, eigene Programme einbinden können, sind alle gängigen Operationen mit den Speicherbau-

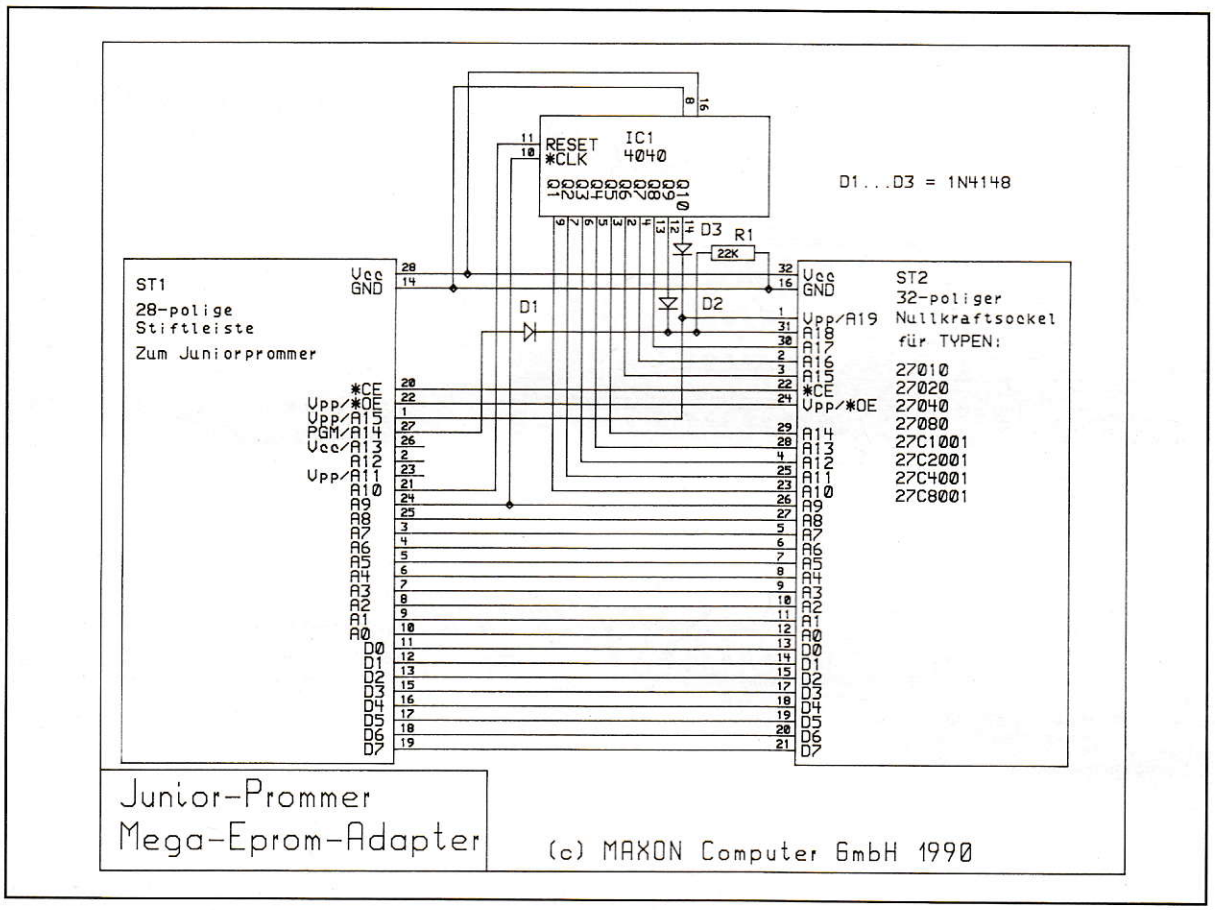
Der Mega-Adapter

Endlich ist es möglich auch 32polige EPROMs mit dem Junior Prommer zu brennen. Benötigt werden dafür der Mega-Adapter und eine angepaßte Software.

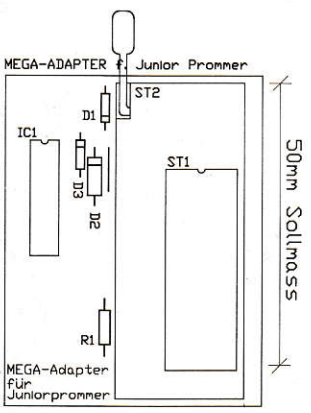
Der Mega-Adapter bereitet die Signale, die am Junior Prommer-Programmiersockel anliegen, auf. Ein Großteil der Leitungen wird einfach durchverbunden (siehe Schaltbild). Die Signale an den Pins 21 und 24 steuern einen C-MOS-Zähler (4040), der die zehn höchsten Adreßleitungen (A10 bis A19) der 32poligen EPROMs ansteuert.

Mit Hilfe dieses Adapters muß während des Brennens kein Schalter zur Adreßbestimmung umgelegt werden. Die zu brennende Datei wird komplett in den Rechner eingelesen, und die Software brennt die 32poligen EPROMs in einem Durchgang.

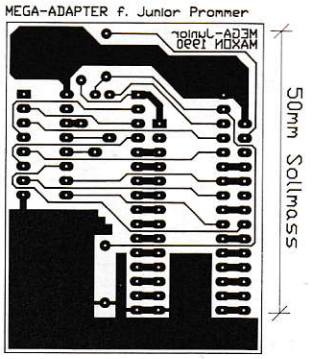
RH/tw



Das Schaltbild zum Mega-Adapter



Der Bestückungsaufdruck



Das Platinen-Layout (Lötseite)

IC1: 4040
ST1: 28polige doppelreihige Stiftleiste
ST2: 32poliger Nullkraftsocke
D1-D3: 1N4148
R1: 22kΩ

Die Stückliste

steinen, wie z.B. Lesen, Programmieren, Vergleichen etc., möglich. Außerdem kann dieses Assembler-Programm durch eine umfassende Konfigurationstabelle an die ausgefallensten Speicherbausteine (auch Banking, z.B. 27011) angepaßt werden. Dieses Assembler-Modul hat den Namen Juniorsoft und wird hier in Verbindung mit einem GFA-BASIC-Programm, genannt BEISPIEL.BAS, vorgestellt. Die gegebenen Beispiele sind generell auch auf andere Hochsprachen übetragbar.

Juniorsoft

Der Assembler-Quellcode von Juniorsoft (Junior.q) ist völlig lageunabhängig und sollte auch ohne TOS-Programmkopf assembliert werden. Junior.q beinhaltet folgenden Funktionen:

1. Initialisieren des Juniorprommers für den weiteren Betrieb.

2. Bedienen der Schieberegister SRA bis SRD.

3. Bedienen des Adreßzählers.

4. Kontrollieren der Datenrichtung.

Patch As Patch Can

Unter diesem Motto stellen wir Ihnen nun ein paar Änderungsvorschläge für das neue Rainbow-TOS vor. Diese Änderungen können jedoch nur vorgenommen werden, wenn Sie das TOS 1.4 vom 6.4.1989 besitzen. Um das Betriebsystem patchen zu können, müssen Sie entweder die ROMs Ihres Rechners mit dem Junior Prommer auslesen, oder, wenn dies nicht möglich ist, mit dem Programm TOS.IMG, welches sich auf der Diskette zum Heft befindet, das Betriebsystem auf Diskette abspeichern. Nun bestehen zwei Möglichkeiten für das weiter Vorgehen. Zum einen können Sie die Änderungen mit einem Diskettenmonitor direkt in dem File auf der Diskette vornehmen. Hierzu dient Tabelle 1. Danach muß das Betriebsystem, um es in sechs EPROMs brennen zu können, mit dem Programm SPLIT aufgeteilt werden.

Die zweite Möglichkeit besteht nun darin, daß Sie das Betriebsystem-File erst aufteilen, und dann die Änderungen in den sechs Teilen vornehmen. Hierzu bedienen Sie sich bitte Tabelle 2. Als letzten Schritt müssen in beiden Fällen die EPROMs gebrannt werden. Dabei, und beim anschließenden Einsetzen in den Rechner achten Sie bitte darauf, daß Sie kein EPROM vertauschen, da Ihr Rechner sonst nicht läuft.

Urs Thürmann/HE

Patch 1

Beim Warmstart wurde bisher das GEMDOS-Datum auf das Erstellungsdatum des TOS (6.4.1989) gestellt und dann geprüft, ob die MEGA-Uhr vorhanden ist. Falls ja, wurde ihr Inhalt ins GEMDOS-Datum und die GEMDOS-Zeit übertragen. Jetzt wird zusätzlich, falls die MEGA-Uhr nicht vorhanden ist, die Uhr des Tastaturprozessors gelesen und - falls sie einen gültigen Wert hat - als GEMDOS-Zeit übernommen. ATARI hat im neuen TOS diese Funktion schon implementiert, durch einen Programmierfehler arbeitete sie jedoch nicht richtig. ST-Besitzer, die keine batteriegepufferte Uhr haben, brauchen die Zeit jetzt nur noch beim Einschalten des Rechners zu stellen.

Abfrage, ob MEGA-Uhr vorhanden

```
FC0448 6A18          bpl.s   $FC0462
```

Änderung in EPROM U4 / HI-0

Patch 2

Der Boot-Device-Fehler ist behoben, so daß der Environment-String richtig gesetzt wird.

_bootdev richtig auslesen

```
FC04B8 3039 ...      move.w  $000446.l,d0
```

Änderung in EPROM U4 / HI-0

Patch 3

Eine falsche Stack-Korrektur in der Routine, die Auto-Ordner-Programme ausführt, konnte unter ungünstigen Umständen zu Abstürzen führen.

Stackpointer korrigieren in autoexec

```
FC0C7C DEFC 000C  add.w   #12,a7
FC0C80 4A40       tst.w   d0
FC0C82 6668       bne.s   $FC0CEC
FC0C84 3F3C 0007  move.w  #7,-(a7)
FC0C88 2F38 0984  move.l  $0984,-(a7)
```

Änderungen in EPROM U4 / HI-0 und U2 / LO-0

Patch 4

Auch der bekannte Fastload-Patch ist hier angegeben, welcher aber mit Vorsicht zu genießen ist. Einige Laufwerke - besonders von NEC - neigen bei ihm zu wesentlich größerer Fehlerhäufigkeit. Ich habe aus diesem Grund diesen Patch nicht in mein EPROM eingebaut, zumal sich fast die gleiche Geschwindigkeit durch speziell formatierte Disketten erreichen läßt (Sektorversatz).

Fastload

```
FC1516 7C10          moveq   #$10,d6
```

Änderung in EPROM U7 / LO-0

Patch 5

Dieser Patch ist dem AUTO-Ordner-Programm von ATARI entnommen und behebt einen Fehler bei der Einstellung der seriellen Schnittstelle.

Rsconf (XBIOS 15)

```
FC3A3E  B07C 0003  cmp.w   #3,d0
FC3A42  6214       bhi.s
$FC3a58
FC3A44  660E       bne.s
$FC3A54
FC3A46  6008       bra.s
$FC3A50
FC3A48  4E71       nop
FC3A4A  4E71       nop
FC3A4C  4E71       nop
FC3A4E  4E71       nop
```

Änderungen in EPROM U4 / HI-0 und U2 / LO-0

Patch 6

Auch dieser Patch stammt aus ATARIs Patch-Programm und behebt einen Fehler des AES bei der Suche des Dateinamens in einer kompletten Pfadangabe.

AES: Dateinamen im Pfad finden (LINE-F: $F5DC)

```
FE411E  226F 0004  move.l  4(a7),a1
FE4122  2449       move.l  a1,a2
FE4124  4A19       tst.b   (a1)+
FE4126  66FC       bne.s   $FE4124
FE4128  1021       move.b  -(a1),d0
FE412A  B3CA       cmp.l   a2,a1
FE412C  650C       bcs.s   $FE413A
FE412E  B03C 005C  cmp.b   #$5C,d0
FE4132  6706       beq.s   $FE413A
FE4134  B03C 003A  cmp.b   #$3A,d0
FE4138  66EE       bne.s   $FE4128
FE413A  5289       addq.l  #1,a1
FE413C  2009       move.l  a1,d0
FE413E  4E75       rts
FE4140  4E71       nop
FE4142  4E71       nop
FE4144  4E71       nop
FE4146  4E71       nop
FE4148  4E71       nop
```

Änderungen in EPROM U2 / HI-2 und U5 / LO-2

Die weiteren Funktionen basieren auf den bisher genannten:

5. Lesen des Speicherbausteins in den Speicher des Rechners.

6. Testen des Speicherbausteins, ob dieser leer ist (nur für EPROMs sinnvoll).

7. Vergleichen des Speicherbausteins mit dem Speicher des Rechners.

8. Prüfen, ob die Daten im Speicher des Rechners problemlos in den Speicherbaustein programmiert wedren können (Nur für EPROMs sinnvoll).

9. Programmieren eines Speichers.

10. Prüfen, ob ein Speicherbaustein des gewünschten Typs eingesetzt ist.

11. Speichertyp einstellen.

12. Initialisieren des Juniorprommers.

Das hier ausschließlich von 'Speicherbausteinen' und nicht von EPROMs geschrieben wird, hat folgenden Grund: Über eine flexible Konfigurationstabelle kann nahezu jeder Speicherbaustein mit 28 oder 24 Anschlüssen über den Junior Prommer bedient werden. So ist es z.B. auch möglich, batteriegepufferte SRAMs zu programmieren, exotische ROMs zu lesen oder EEPROMs zu programmieren.

In der Typentabelle, sind bereits die EPROMs von 2716 bis 27011 enthalten.

Patch As Patch Can

Patch	Adresse	alt	neu
(1)	$448	$6418	$6A18
(2)	$4B8	$1039	$3039
(3)	$C7C	$5C4F	$DEFC
		$4A40	$000C
		$666A	$4A40
		$3F3C	$6668
		$0007	$3F3C
		$2F39	$0007
		$0000	$2F38
(4)	$1516	$7C14	$7C10
(5)	$3A40	$FFFF	$0003
		$6714	$6214
		$1140	$660E
		$0020	$6008
		$670A	$4E71
		$C03C	$4E71
		$00FD	$4E71
		$6704	$4E71
(6)	$2411E	$4E56	$226F
		$0000	$0004
		$48E7	$2449
		$0104	$4A19
		$2A6E	$66FC
		$0008	$1021
		$4A1D	$B3CA
		$66FC	$650C
		$6002	$B03C
		$538D	$005C
		$BBEE	$6706
		$0008	$B03C
		$650C	$003A
		$0C15	$66EE
		$005C	$5289
		$6706	$2009
		$0C15	$4E75
		$003A	$4E71
		$66EC	$4E71
		$528D	$4E71
		$200D	$4E71
		$F801	$4E71

Tabelle 1: Nach Methode 1 wird die Datei TOS.IMG mittels eines Diskmonitors verändert und dann erst aufgesplittet.

Patch	EPROM	Adresse	alt	neu
(1)	HI-0	224	6461	6A61
(2)	HI-0	25C	1000	3000
(3)	HI-0	63E	5C4A	DE00
			663F	4A66
			002F	3F00
			0009	2F09
	LO-0	63E	4F40	FC0C
			6A3C	4068
			0739	3C07
			0084	3884
(4)	LO-0	A8B	1424	1024
(5)	HI-0	1D20	FF67	0062
			1100	6660
			67C0	4E4E
			0067	4E4E
	LO-0	1D20	FF14	0314
			4020	0E08
			0A3C	7171
			FD04	7171
(6)	HI-2	208F	4E00	2200
			4801	244A
			2A00	6610
			4A66	B365
			6053	B000
			BB00	67B0
			650C	0066
			0067	5220
			0C00	4E4E
			6652	4E4E
			20F8	4E4E
	L0-2	208F	5600	6F04
			E704	4919
			6E08	FC21
			1DFC	CA0C
			028D	3C5C
			EE08	063C
			0C15	3AEE
			5C06	8909
			153A	7571
			EC8D	7171
			0D01	7171

Tabelle 2: Nach Methode 2 werden die Änderungen erst nach dem Aufsplitten des TOS in sechs Teile vorgenommen.

Wie Sie neue Speichertypen definieren können, wird weiter unten erläutert. Es liegt in Ihrem Ermessen, diese Typentabelle beliebig zu erweitern.

Handhabung der Konfigurationstabelle in der Praxis

Falls Sie einen neuen Typ implementieren wollen, sollten Sie zuerst prüfen, ob dieser überhaupt geeignet ist. Die Steuereleitungen und restlichen Adreßleitungen sollten von SRA oder SRB abgedeckt werden und die Adreß-, Daten-, Versorgungs- und Programmierspannungsleitungen, welche bereits fest verdrahtet sind, sollten an den festgelegten Pins liegen. Zu beachten ist, daß Speicher mit bis zu 17 Adreßleitungen möglich sind. Nun füllen Sie am besten die Konfigurationstabelle mit Hilfe eines Datenblattes des Speichertyps aus. Einträge in der Konfigurationstabelle, die von Juniorsoft nicht explizit genutzt werden, können Sie von Ihren eigenen Programmen aus belegen. Zuletzt sollten Sie nicht vergessen, die LED auch in den verschiedenen Kontrollzuständen des EPROMs zu berücksichtigen. Die Konfigurationstabelle hat 82 Bytes pro Eintrag (Typ; siehe Kasten 'Die Typenkonfigurationstabelle'). Sie sollten immer darauf achten, daß genug freier Speicher für die Konfigurationstabelle vorhanden ist. Wohin Sie die Konfigurationstabelle im Arbeitsspeicher ablegen, muß immer den Gegebenheiten angepaßt werden. In GFA-BASIC bietet sich z.B. die Möglichkeit an, diese in einer Zeichenkette abzulegen, die vorher auf die geforderte Länge mit Leerzeichen aufgefüllt worden ist.

Die Juniorsoft-Funktionen in der Praxis

Eine Beispielanwendung der Juniorsoft-Funktionen ist im Programm JUNIOR auf der Diskette zum Heft gezeigt. Dises dient zum Testen und Kalibrieren (Spannungseinstellung an TR1, neben IC7 - TL497).

Wie schon geschrieben, dienen die Funktionen 0 bis 3 zur Ansteuerung des Junior Prommers auf niedrigster Ebene. Die Funktionen 4 bis 11 bauen auf diesen auf und ermöglichen die Ansteuerung des Junior Prommers wie die eines Massenspeichers, benötigen zum Betrieb jedoch die Typentabelle. Es muß immer vor dem Aufruf dieser Funktionen ein gültiger Typ eingestellt werden, da es sonst zu Abstürzen kommen kann.

Die meisten Funktionen, wie z.B. Lesen, Programmieren, benötigen einen Datenspeicher im Rechner. Auch dieser kann z.B. in einer Zeichenkette gehalten werden, jedoch nur bis 32768 Bytes.

Alle Adreßparameter der Funktionen werden nicht auf Gültigkeit geprüft, denn das liegt in dem Aufgabenbereich des Ansteuerungsprogramms. Außerdem sollten Adreßvariablen ausschließlich vom Typ Integer sein.

Das Programm BEISPIEL zeigt, anhand des Kopierens eines EPROMs vom Typ 27128, wie fast alle Funktionen benutzt werden.

Zum Schluß

Wenn Sie nur Ihr Betriebsystem modifizieren möchten (siehe Kasten 'Rund um die Betriebsystem-EPROMs') und dazu nicht erst zum Lötkolben greifen wollen, besteht für Sie die Möglichkeit, ein Fertiggerät (Bild 1) mit komfortabler GEM-gesteuerter Brennsoftware bei der MAXON Computer GmbH zu kaufen. Im Angebot ist dort auch die Leerplatine mit der GEM-Software und einem ausführlichem Handbuch.

Zum Schluß wünschen wir Ihnen viel Spaß beim Vordringen in des Gebiet der selbstgebrannten Speicherbausteine.

RH

EASY RIDER

FÜR ALLE ATARI ST

EASY RIDER - Der intelligente Re- und Disassembler - V2.3

Automatische Unterscheidung von Daten- und Codebereichen, aber manuell beinflußbar, Wandlung von Adressregisteroffsets, Datenbereichen oder Befehlskonstanten in editierfähige Ausdrücke, z.B. statt 6000(A6) -› Ziel1-Base6(A6) oder statt DC.W 400 -› DC.W Ziel2-Bezug, Einbindung von Symboltabellen, Editierung von Symbolen, zahlreiche Suchfunktionen, auch ROM/RAM und beliebige Sektoren werden geladen und alles über Maus und Tastatur erreichbar! Der Reassembler erstellt von jedem Programm einen editier- und assemblierfähigen Quelltext! DM 149,-

EASY RIDER - Der superschnelle Assembler für Profis - V2.0

Macros mit bis zu 99 Parametern, bedingte Assemblierung, Linker, Bibliotheksmanager, Include-Dateien - z. B. Einbindung von DR-/GST-Objektdateien und DR- Objektbibliotheken in ganz normalen Quelltext, beliebig viele lokale Label, intelligente Wortjustierung, Codeoptimierung - auch vorwärts (!) u. v. a. m..

Der Assembler ist natürlich voll kompatibel zum Reassembler - und schnell - schnell - schnell!

Jetzt inclusive TEMPUS V1.11 von CCD! DM 99,-

Paketpreis - Reassembler plus Assembler - nur DM 229,-

erhältlich im guten Fachhandel oder direkt bei A. Borchard
Wiesenbachstr. 2a 4500 Osnabrück Tel.: 0541/87024

EDISON EDITOR

Integrierte Entwicklungs- /Programmierumgebung für alles, was mit Textbearbeitung zu tun hat (Programmierer, Vielschreiber, TEX Benutzer etc). EDISON ist ein schneller, komfortabler Fullscreen GEM Editor und eine frei konfigurierbare Shell mit integrierter Batch-Verarbeitung eingabautem MAKE, Makrorecorder etc. Der Leistungsumfang von EDISON in Stichworten: ■ bis zu 6 GEM Fenster gleichzeitig ■ schnelles Scrolling und Echtzeitscrolling ! ■ verarbeitet alle Zeichen von 0 bis 255, Zeilen bis 32000 Zeichen ■ echte Tabulatoren, sowie einstellbare Tabulatorzeichen (z.B. ';') ■ automatische Anpassung des Lineals an den Quelltext (z.B. Assemblerlistings, Datenbanken) ■ vollautomatische Fehlersuche einfach mit HELP (an praktisch alle Compiler anpassbar !) ■ Wort-, Zeilen-, Klammerblock einfach durch Doppelklick ■ sehr flexible und kinderleichte MAKRO-Programmierung (Ereignisrecorder, auch Menüs und Dialogboxen dürfen aufgerufen werden ! ■ frei konfigurierbare eingebaute Shell mit Batch Verarbeitung (bis zu 40 !! Batchjobs gleichzeitig ■ ca. 20 eingebaute Shellkommandos (z.B. Rename, Chdir, Del, Input, Copy, Output, Stack, ...) ■ integriertes Make (EDISON compiliert nur die Files, die geändert wurden) ■ läuft auf allen ATARI ST, TT (Monochrom-, Farb- und Großbildschirm) ■ Der Leistungsumfang von EDISON ist so umfangreich, daß er hier nur angedeutet werden kann. Forden Sie deshalb die Demodiskette (mit Anleitung auf Diskette) an!

EDISON Demodiskette gegen DM 10.- (Schein, Scheck, Briefmarken)

EDISON komplett mit deutschem Handbuch DM 169,-

KNISSOFT

hans christian kniß adalbertstraße 44 d-5100 aachen tel: 0241/24252

Platon Leiterplatten CAD-System für Atari ST

- Vektororientiertes Programm, dadurch hohe Speichereffizienz
- Maximale Bearbeitungsgröße 832x832 mm, 8 Lagen
- Auflösung bis zu 1/320 Zoll (intern 1/2000 Zoll)
- Lauffähig auf Großmonitor
- Umfangreiche Bauteilebibliothek, Bauteileverwaltung
- Zoom, vorbildliche Textfunktionen, Gummibandtechnik, einlesen verschiedener Dateiformate, Millimeterraster u.v.a.
- Ausgabe auf Drucker, Plotter, XYZ-Anlage, Fotoplotter u.a. Ausgabe in Bohrdaten-, Gerber-, HP-GL-Datei
- Bohren von Leiterplatten, Fräsen von Frontplatten o.ä.
- Lieferung inkl. Profi-Dateiauswahlfenster
- siehe Test in ST-Computer 3/90 (seitdem weitere Verbesserungen)

Platon V 1.4
Demoversion (wird vergütet) 20,-- DM
zzgl. 5,70 DM Versandpauschale
Weitere Infos (Programmprofil) Gratis
298,-- DM

VHF Computer
Vogt, Henne, Fleischmann GbR
Maurener Weg 115a
7030 Böblingen
07031/289211

+5V
Gnd
-5V
74HC00
4,7k

Diese Anzeige wurde vollständig mit Platon erstellt